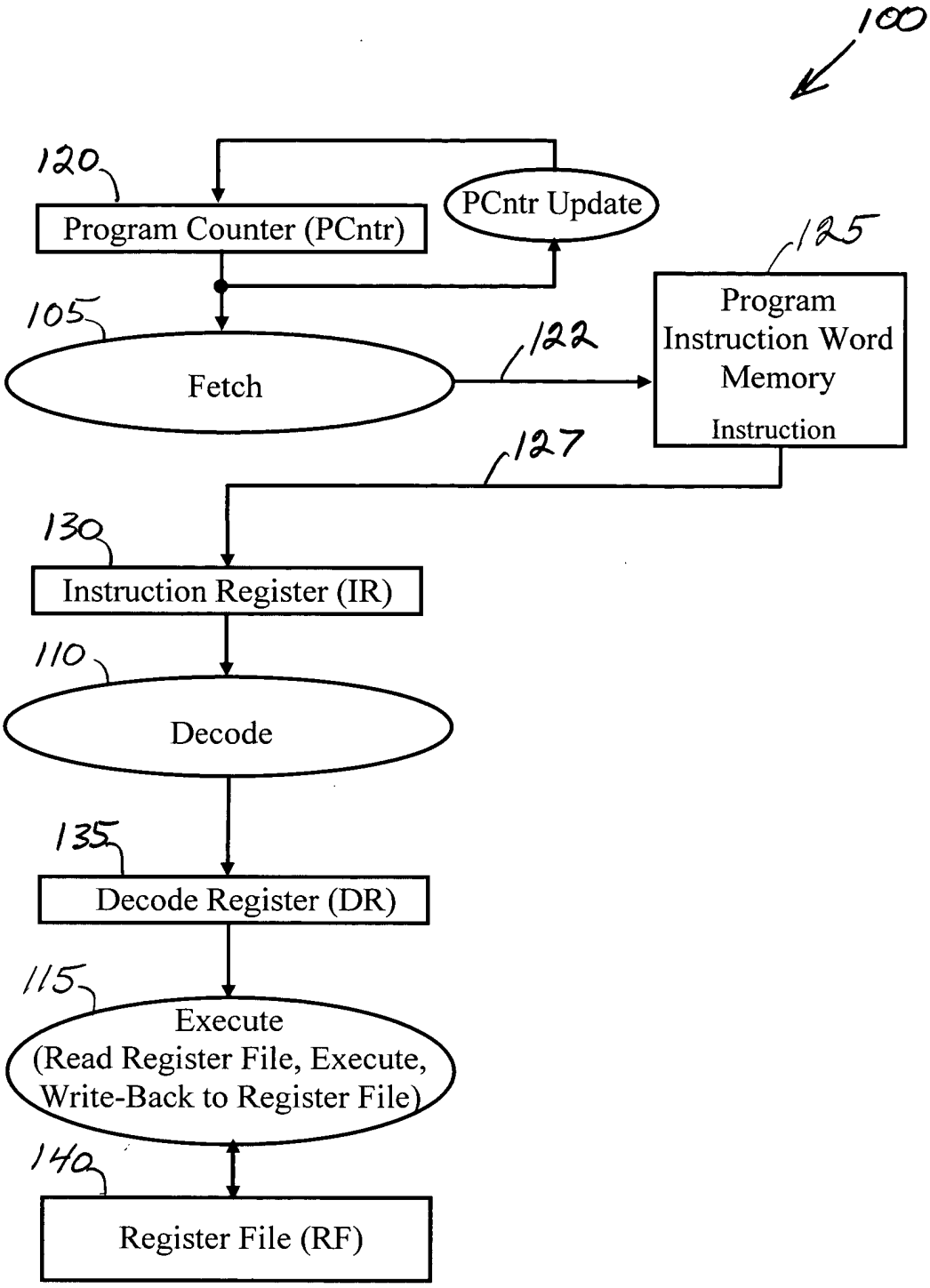

100
120
Program Counter (PCntr)
PCntr Update
125
Program Instruction Word Memory
105
Fetch
122
Instruction
127
130
Instruction Register (IR)
110
Decode
135
Decode Register (DR)
115
Execute
(Read Register File, Execute, Write-Back to Register File)
140
Register File (RF)

Fig. 1

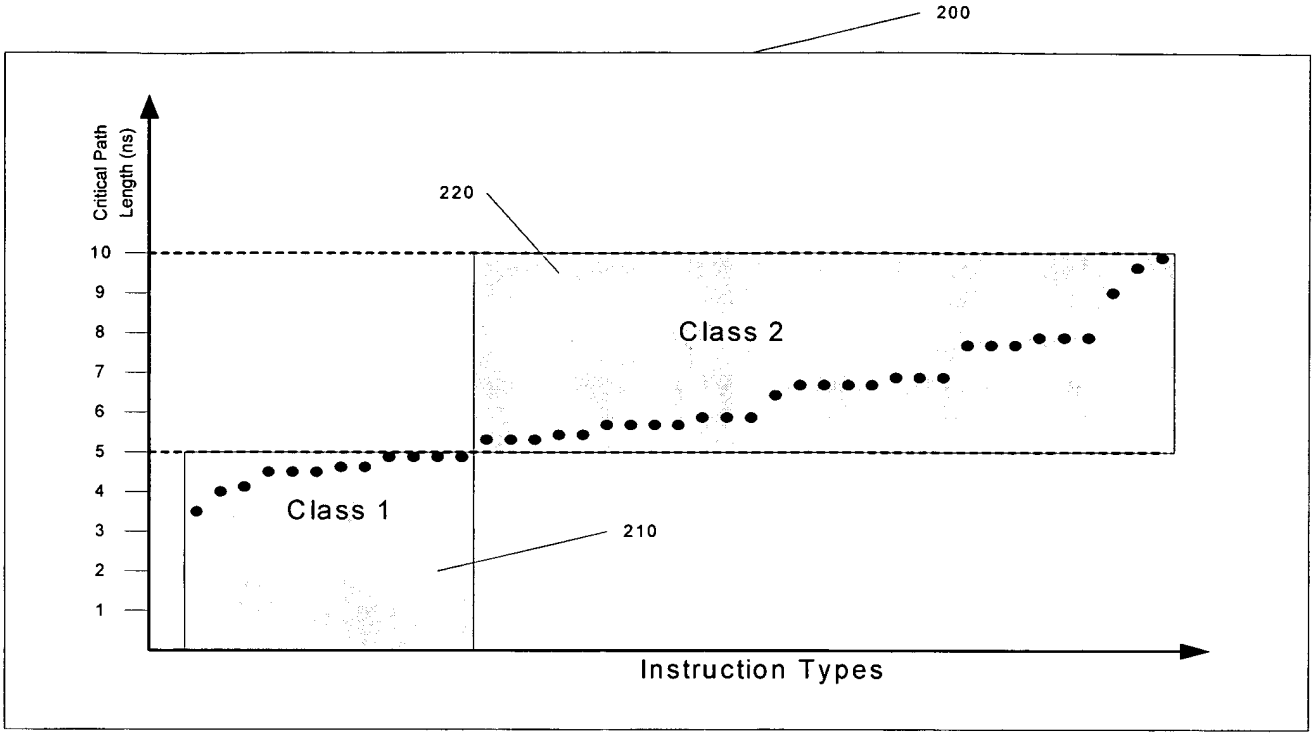

200
Critical Path Length (ns)
10
9
8
7
6
5
4
3
2
1
220
Class 2
Class 1
210
Instruction Types

Fig. 2

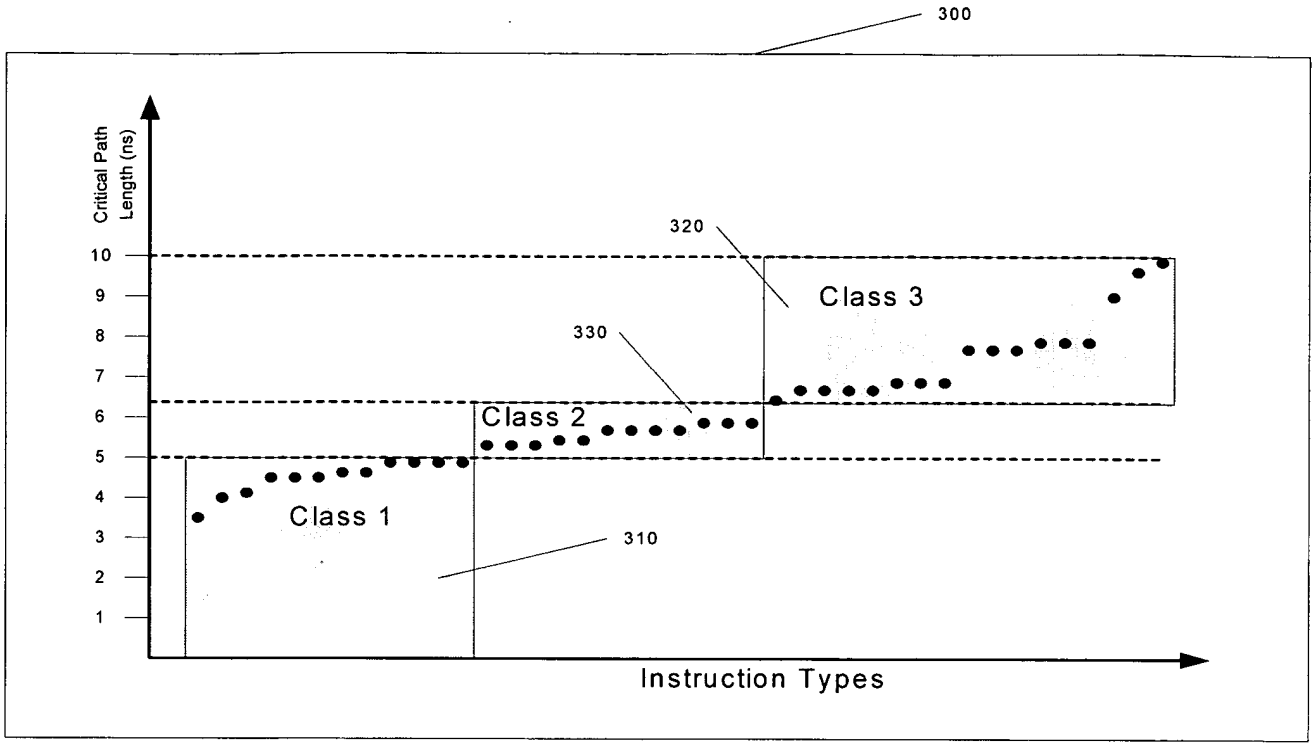

300
Critical Path Length (ns)
10
9
8
7
6
5
4
3
2
1
320
Class 3
330
Class 2
Class 1
310
Instruction Types

Fig. 3

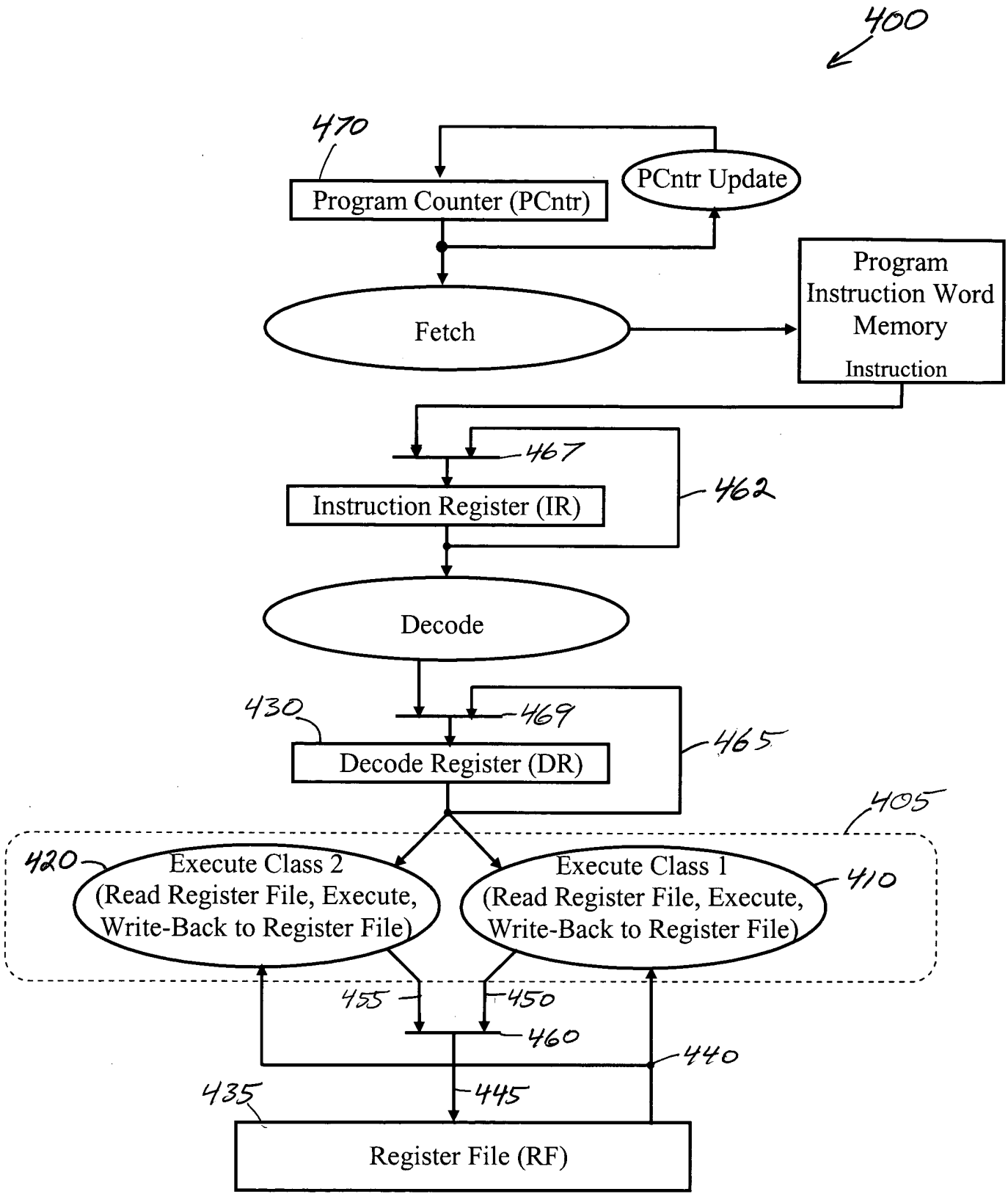
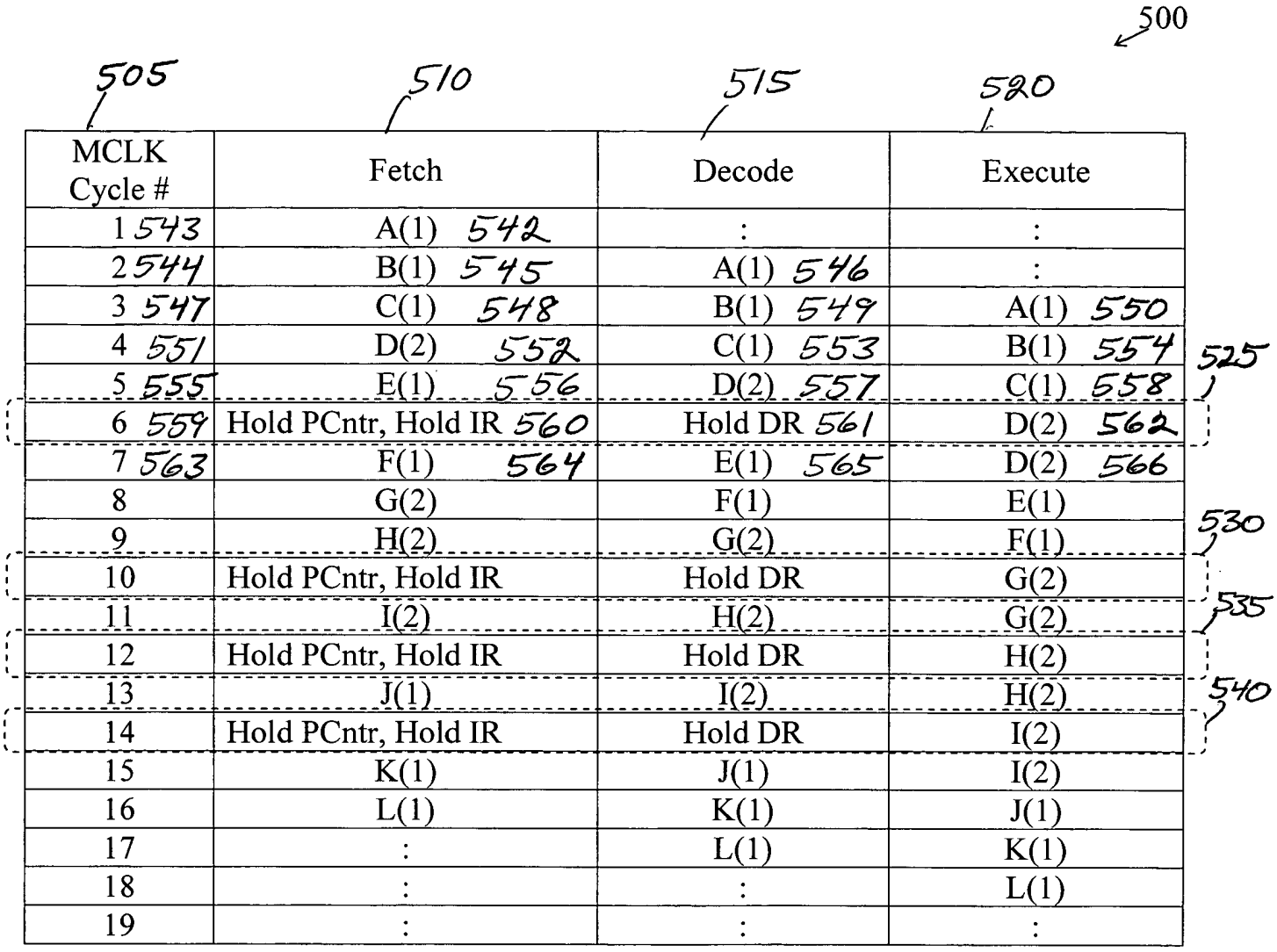

400
470
Program Counter (PCntr)
PCntr Update
Fetch
Program Instruction Word Memory
Instruction
467
462
Instruction Register (IR)
Decode
430
469
465
Decode Register (DR)
405
420
Execute Class 2 (Read Register File, Execute, Write-Back to Register File)
Execute Class 1 (Read Register File, Execute, Write-Back to Register File)
410
455
450
460
440
435
445
Register File (RF)

Fig. 4



MCLK Cycle #	Fetch	Decode	Execute
1 543	A(1) 542	:	:
2 544	B(1) 545	A(1) 546	:
3 547	C(1) 548	B(1) 549	A(1) 550
4 551	D(2) 552	C(1) 553	B(1) 554
5 555	E(1) 556	D(2) 557	C(1) 558
6 559	Hold PCntr, Hold IR 560	Hold DR 561	D(2) 562
7 563	F(1) 564	E(1) 565	D(2) 566
8	G(2)	F(1)	E(1)
9	H(2)	G(2)	F(1)
10	Hold PCntr, Hold IR	Hold DR	G(2)
11	I(2)	H(2)	G(2)
12	Hold PCntr, Hold IR	Hold DR	H(2)
13	J(1)	I(2)	H(2)
14	Hold PCntr, Hold IR	Hold DR	I(2)
15	K(1)	J(1)	I(2)
16	L(1)	K(1)	J(1)
17	:	L(1)	K(1)
18	:	:	L(1)
19	:	:	:

Fig. 5

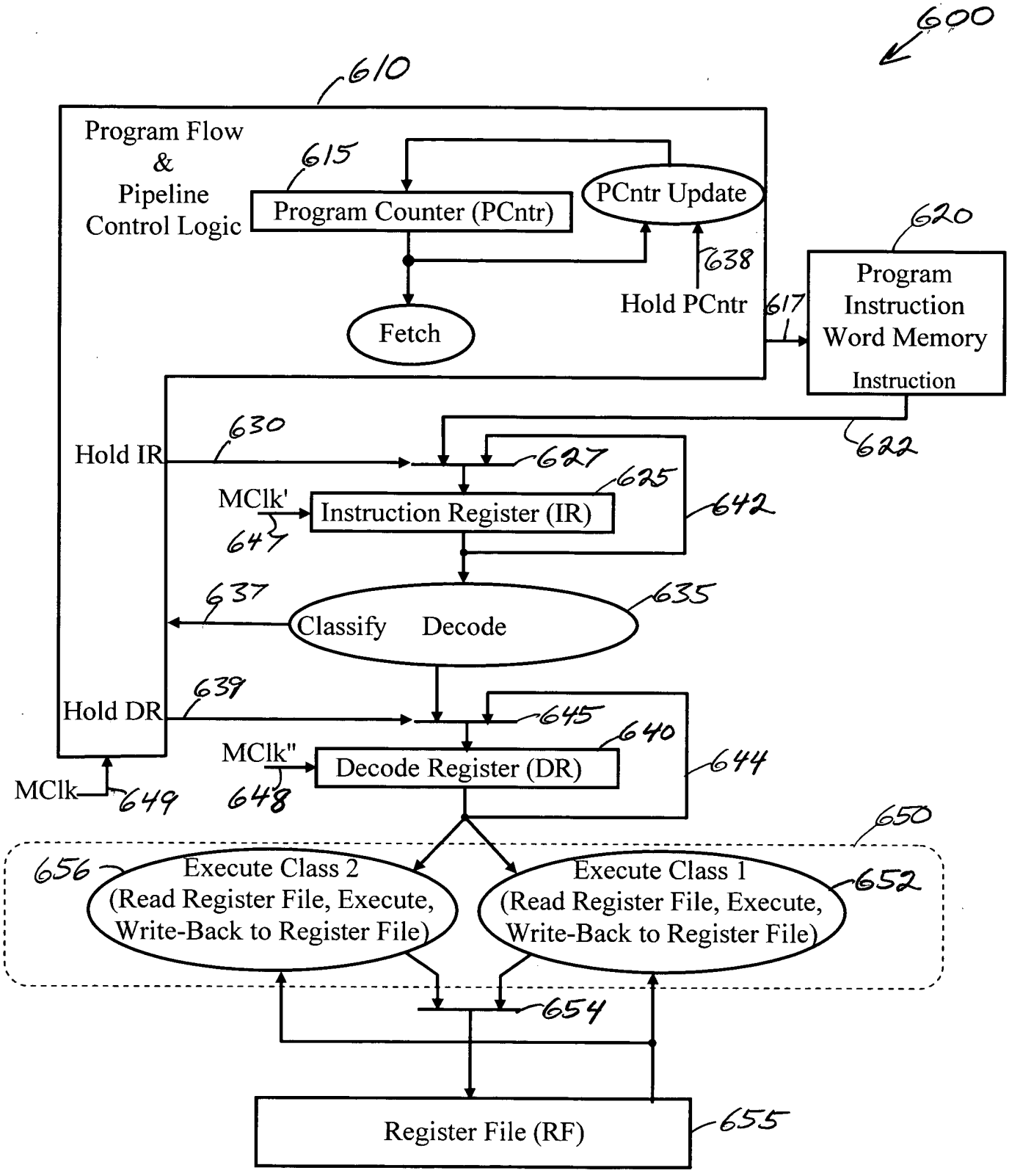

600
610
Program Flow
&
Pipeline
Control Logic
615
Program Counter (PCntr)
PCntr Update
638
Hold PCntr
Fetch
620
Program
Instruction
Word Memory
Instruction
617
622
Hold IR
630
627
625
MClk'
647
Instruction Register (IR)
642
637
Classify Decode
635
Hold DR
639
645
640
MClk"
648
Decode Register (DR)
644
MClk
649
650
656
Execute Class 2
(Read Register File, Execute,
Write-Back to Register File)
Execute Class 1
(Read Register File, Execute,
Write-Back to Register File)
652
654
Register File (RF)
655

Fig. 6

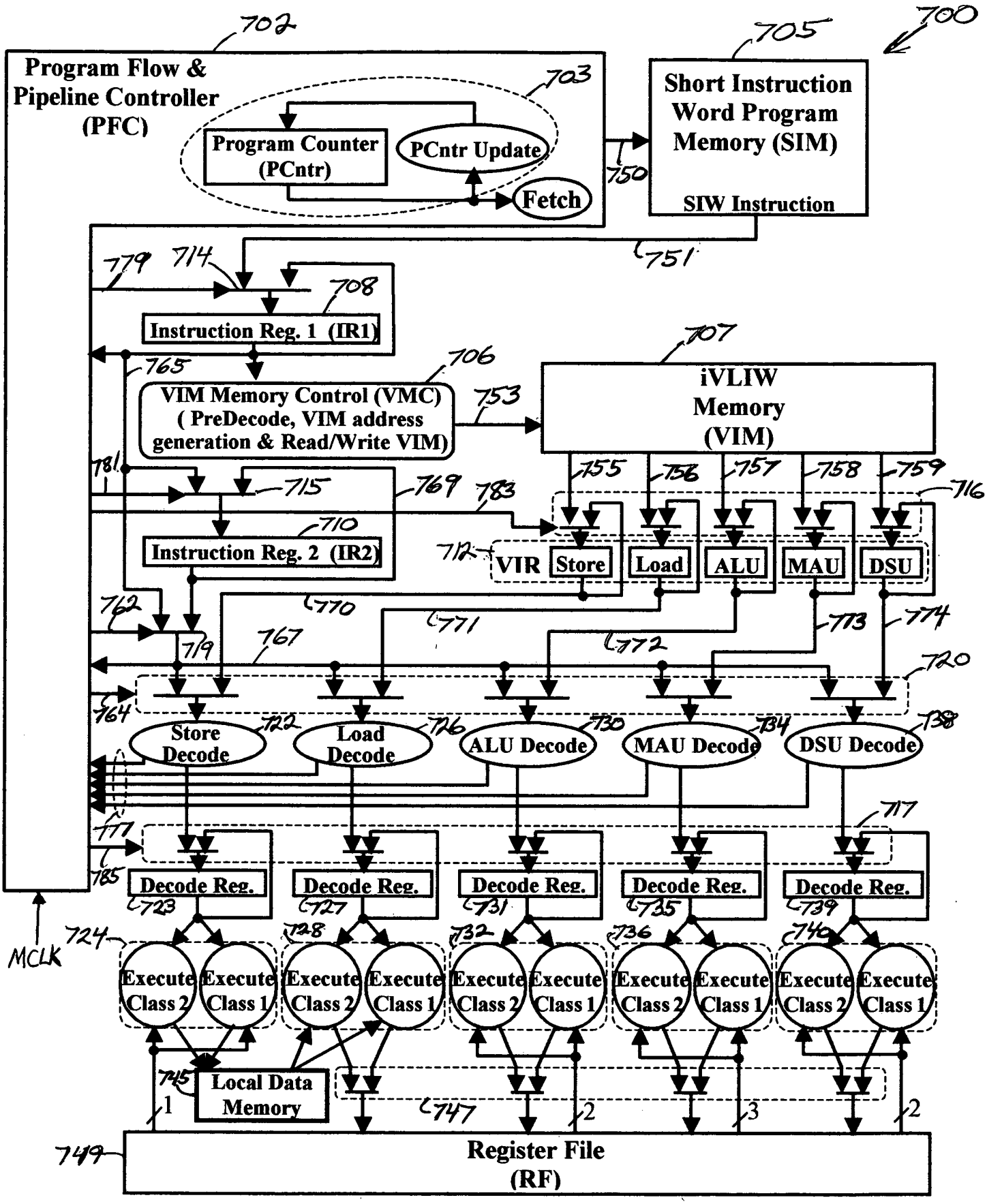

700
702
Program Flow & Pipeline Controller (PFC)
703
Program Counter (PCntr)
PCntr Update
Fetch
705
Short Instruction Word Program Memory (SIM)
SIW Instruction
750
751
779
714
708
Instruction Reg. 1 (IR1)
765
706
VIM Memory Control (VMC) (PreDecode, VIM address generation & Read/Write VIM)
753
707
iVLIW Memory (VIM)
781
715
769
783
710
Instruction Reg. 2 (IR2)
712
VIR
Store
Load
ALU
MAU
DSU
716
755
756
757
758
759
762
770
771
772
773
774
719
767
720
764
Store Decode
722
Load Decode
726
ALU Decode
730
MAU Decode
734
DSU Decode
738
777
717
785
Decode Reg.
723
727
731
735
739
MCLK
724
728
732
736
740
Execute Class 2
Execute Class 1
745
Local Data Memory
747
1
2
3
749
Register File (RF)

Fig. 7

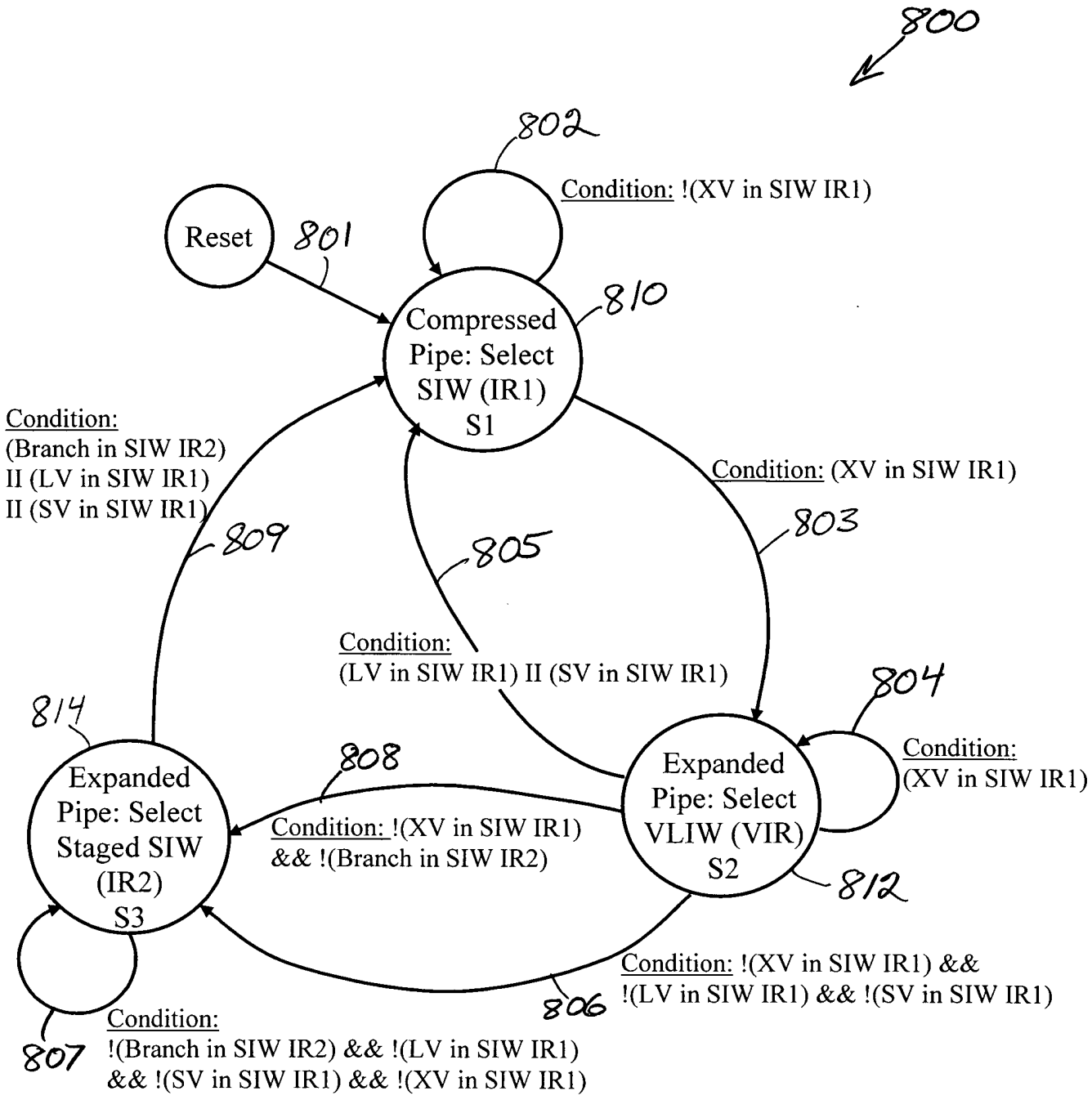
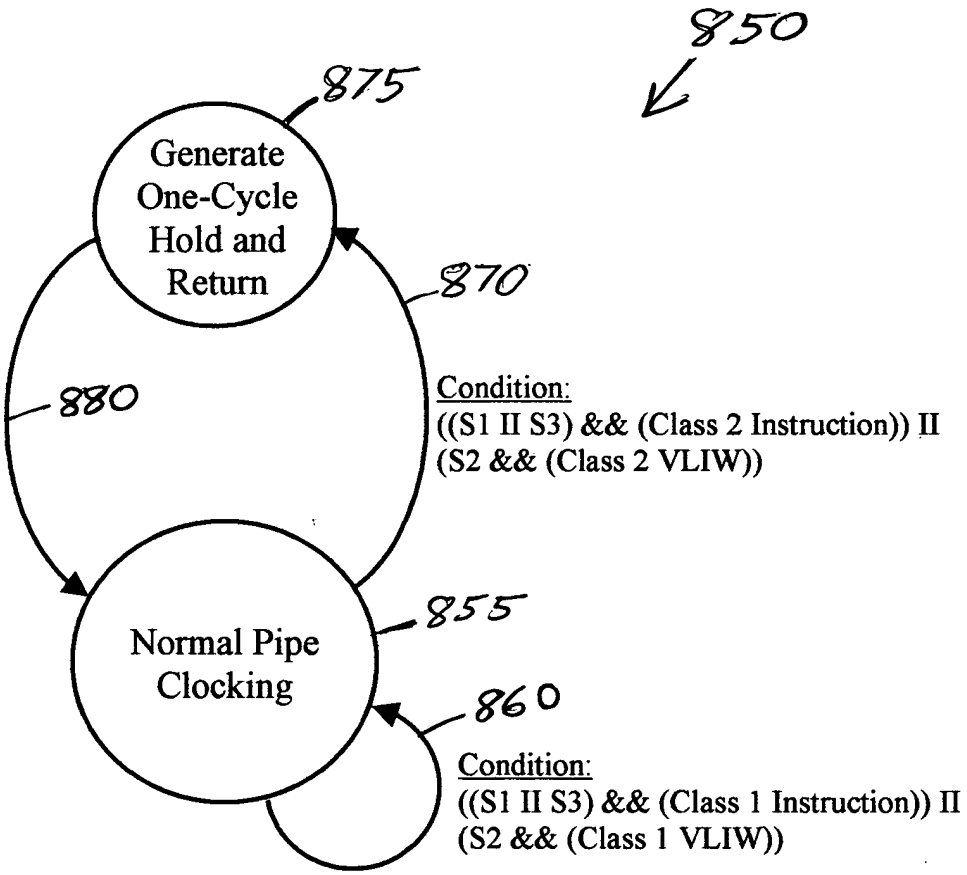

800
802
Condition: !(XV in SIW IR1)
Reset
801
Compressed Pipe: Select SIW (IR1) S1
810
Condition: (Branch in SIW IR2) II (LV in SIW IR1) II (SV in SIW IR1)
809
Condition: (XV in SIW IR1)
803
805
Condition: (LV in SIW IR1) II (SV in SIW IR1)
804
814
Condition: (XV in SIW IR1)
808
Expanded Pipe: Select Staged SIW (IR2) S3
Condition: !(XV in SIW IR1) && !(Branch in SIW IR2)
Expanded Pipe: Select VLIW (VIR) S2
812
Condition: !(XV in SIW IR1) && !(LV in SIW IR1) && !(SV in SIW IR1)
806
Condition: !(Branch in SIW IR2) && !(LV in SIW IR1) && !(SV in SIW IR1) && !(XV in SIW IR1)
807

Fig. 8A



S1 = Compressed Pipe: Select SIW (IR1)
S2 = Expanded Pipe: Select VLIW (VIR)
S3 = Expanded Pipe: Select Staged SIW (IR2)
Class 1 VLIW = All Enabled VLIW Slot Instructions are Class 1
Class 2 VLIW = One or More Enabled VLIW Slot Instructions are Class 2

Fig. 8B

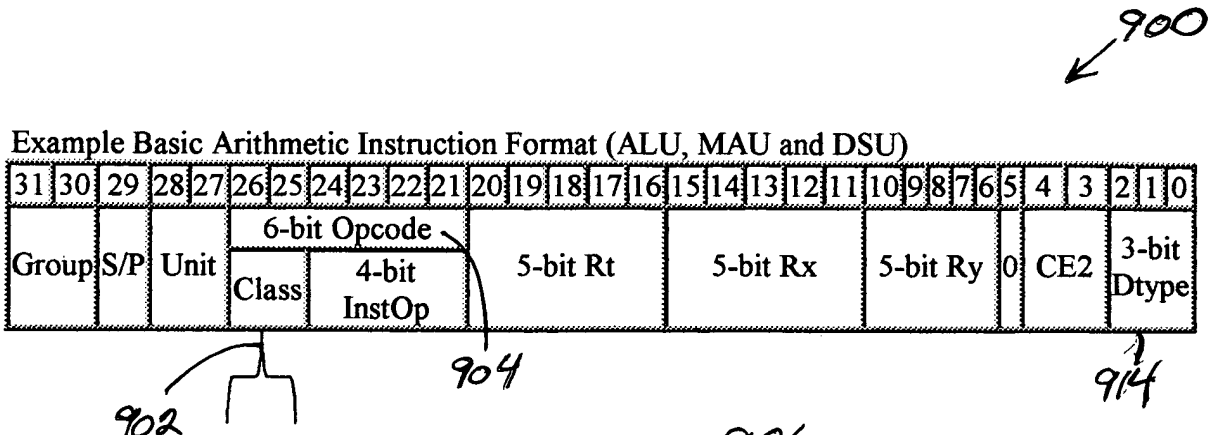

900
Example Basic Arithmetic Instruction Format (ALU, MAU and DSU)
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
S/P
Unit
6-bit Opcode
Class
4-bit InstOp
5-bit Rt
5-bit Rx
5-bit Ry
0
CE2
3-bit Dtype
902
904
914
0 0 = Class 1, PCLK = 3 MCLKs — 906
0 1 = Class 2, PCLK = 4 MCLKs — 908
1 0 = Class 3, PCLK = 6 MCLKs — 910
1 1 = Reserved — 912

Fig. 9A

925

926 MCLK Cycle #	927 PCLK Cycle #	928 Fetch	929 Decode	930 Execute
1 934				
2 935	1 933	A(1) 932	:	:
3 936				
4 940				
5 941	2 937	B(2) 938	A(1) 939	:
6 942				
7 947				
8 948	3 943	C(3) 944	B(2) 945	A(1) 946
9 949				
10 954				
11 955	4 950	D(1) 951	C(3) 952	B(2) 953
12 956				
13 957				
14 962				
15 963				
16 964	5 958	E(1) 959	D(1) 960	C(3) 961
17 965				
18 966				
19 967				
20 972				
21 973	6 968	F(1) 969	E(1) 970	D(1) 971
22 974				
23				
24	7	:	:	:
25				

Fig. 9B

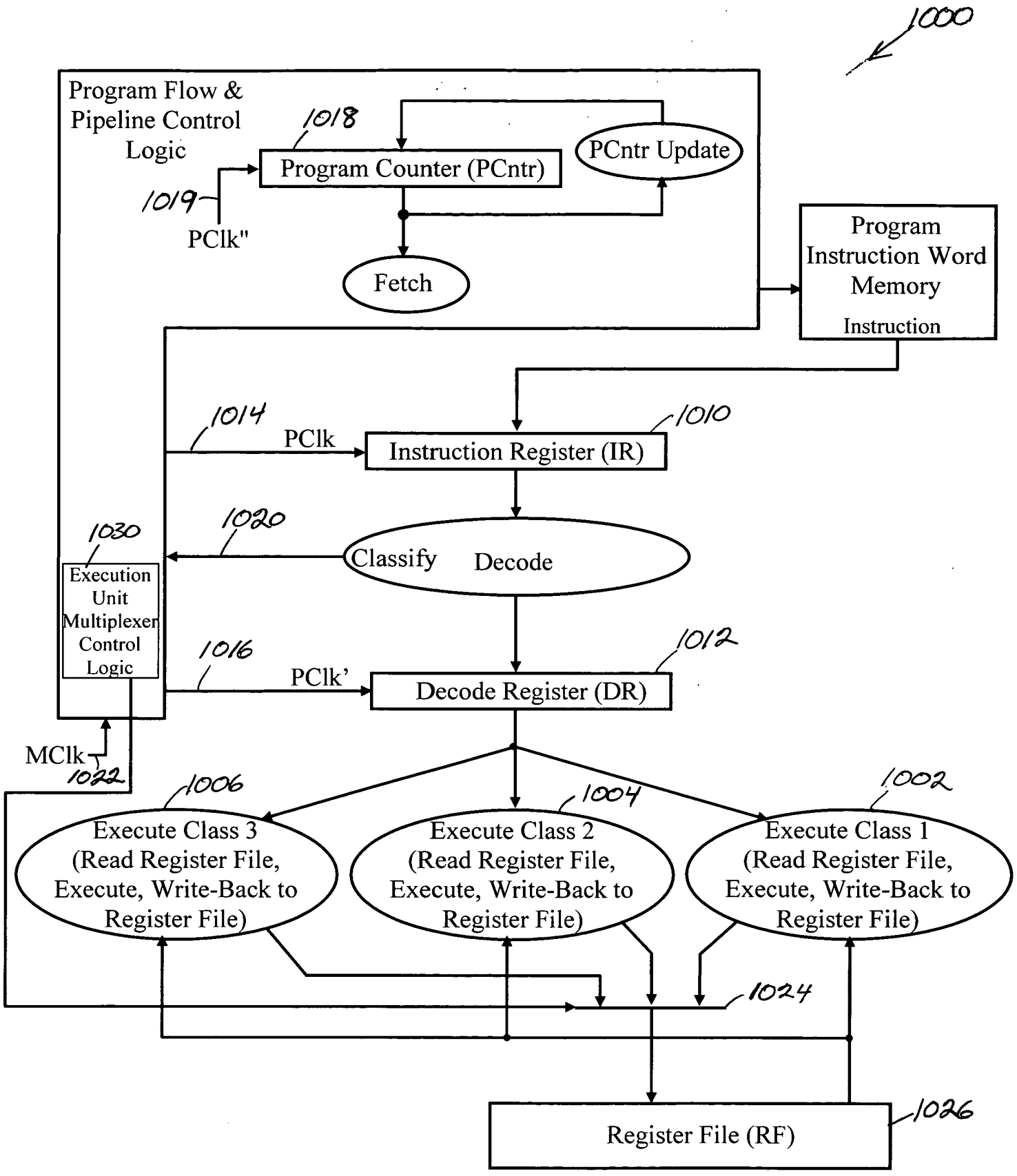

1000
Program Flow & Pipeline Control Logic
1018
Program Counter (PCntr)
PCntr Update
1019
PClk"
Fetch
Program Instruction Word Memory
Instruction
1014
PClk
Instruction Register (IR)
1010
1020
Classify
Decode
1030
Execution Unit Multiplexer Control Logic
1016
PClk'
Decode Register (DR)
1012
MClk
1022
1006
1004
1002
Execute Class 3 (Read Register File, Execute, Write-Back to Register File)
Execute Class 2 (Read Register File, Execute, Write-Back to Register File)
Execute Class 1 (Read Register File, Execute, Write-Back to Register File)
1024
Register File (RF)
1026

Fig. 10

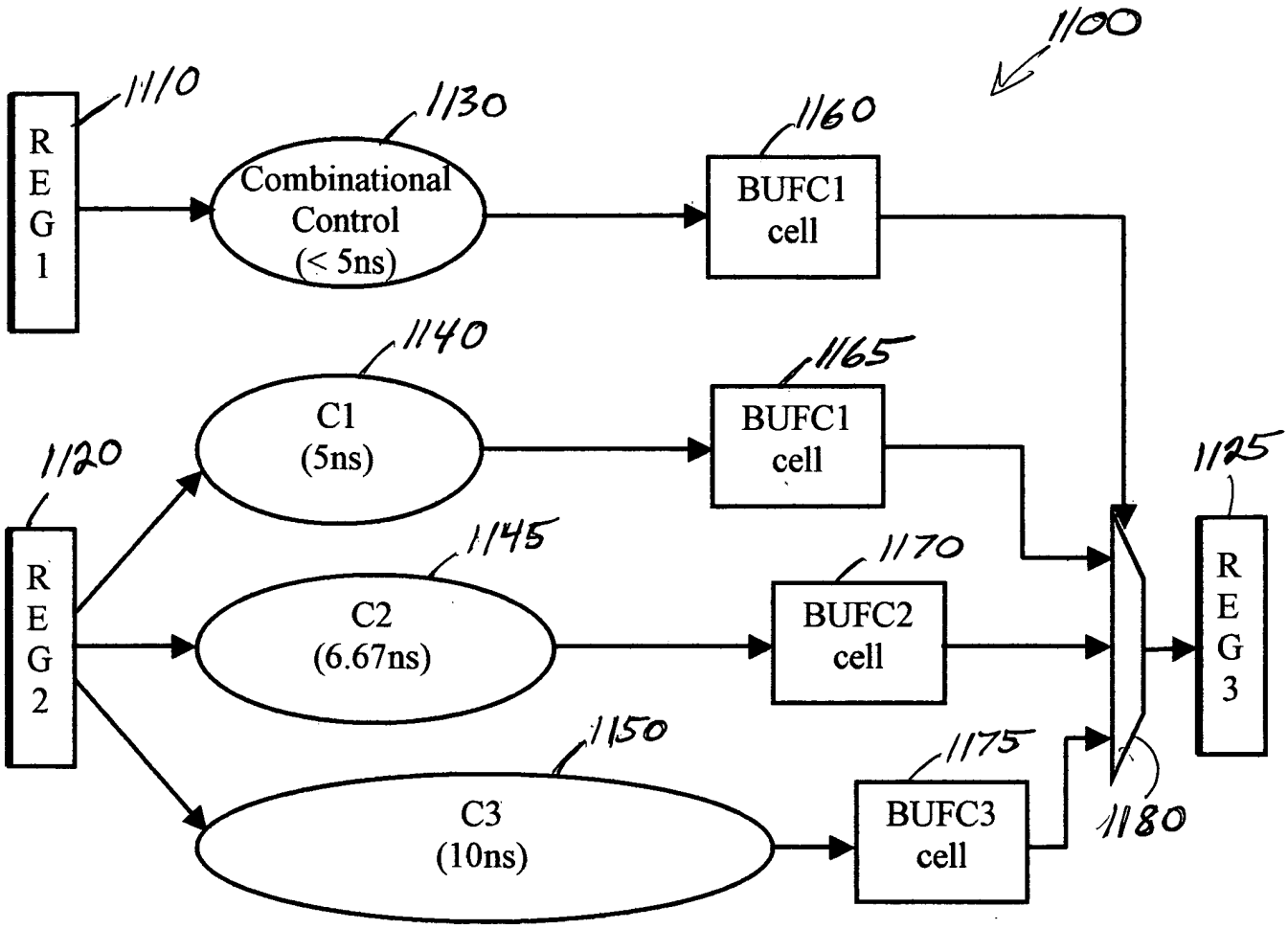

1100
1110
REG 1
1130
Combinational Control (< 5ns)
1160
BUFC1 cell
1140
C1 (5ns)
1165
BUFC1 cell
1120
REG 2
1145
C2 (6.67ns)
1170
BUFC2 cell
1125
REG 3
1150
C3 (10ns)
1175
BUFC3 cell
1180

Fig. 11

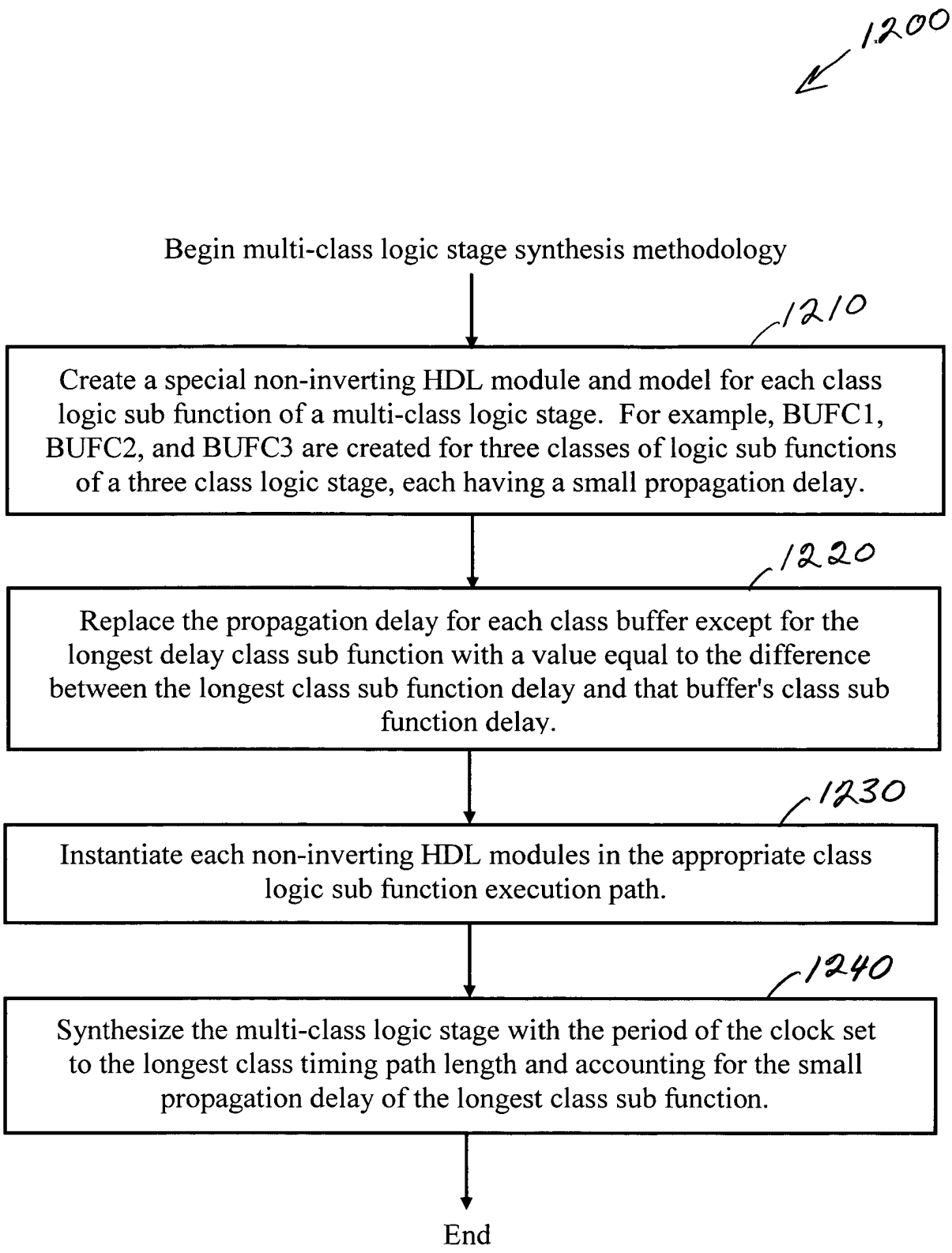

1200
Begin multi-class logic stage synthesis methodology
1210
Create a special non-inverting HDL module and model for each class logic sub function of a multi-class logic stage. For example, BUFC1, BUFC2, and BUFC3 are created for three classes of logic sub functions of a three class logic stage, each having a small propagation delay.
1220
Replace the propagation delay for each class buffer except for the longest delay class sub function with a value equal to the difference between the longest class sub function delay and that buffer's class sub function delay.
1230
Instantiate each non-inverting HDL modules in the appropriate class logic sub function execution path.
1240
Synthesize the multi-class logic stage with the period of the clock set to the longest class timing path length and accounting for the small propagation delay of the longest class sub function.
End

Fig. 12

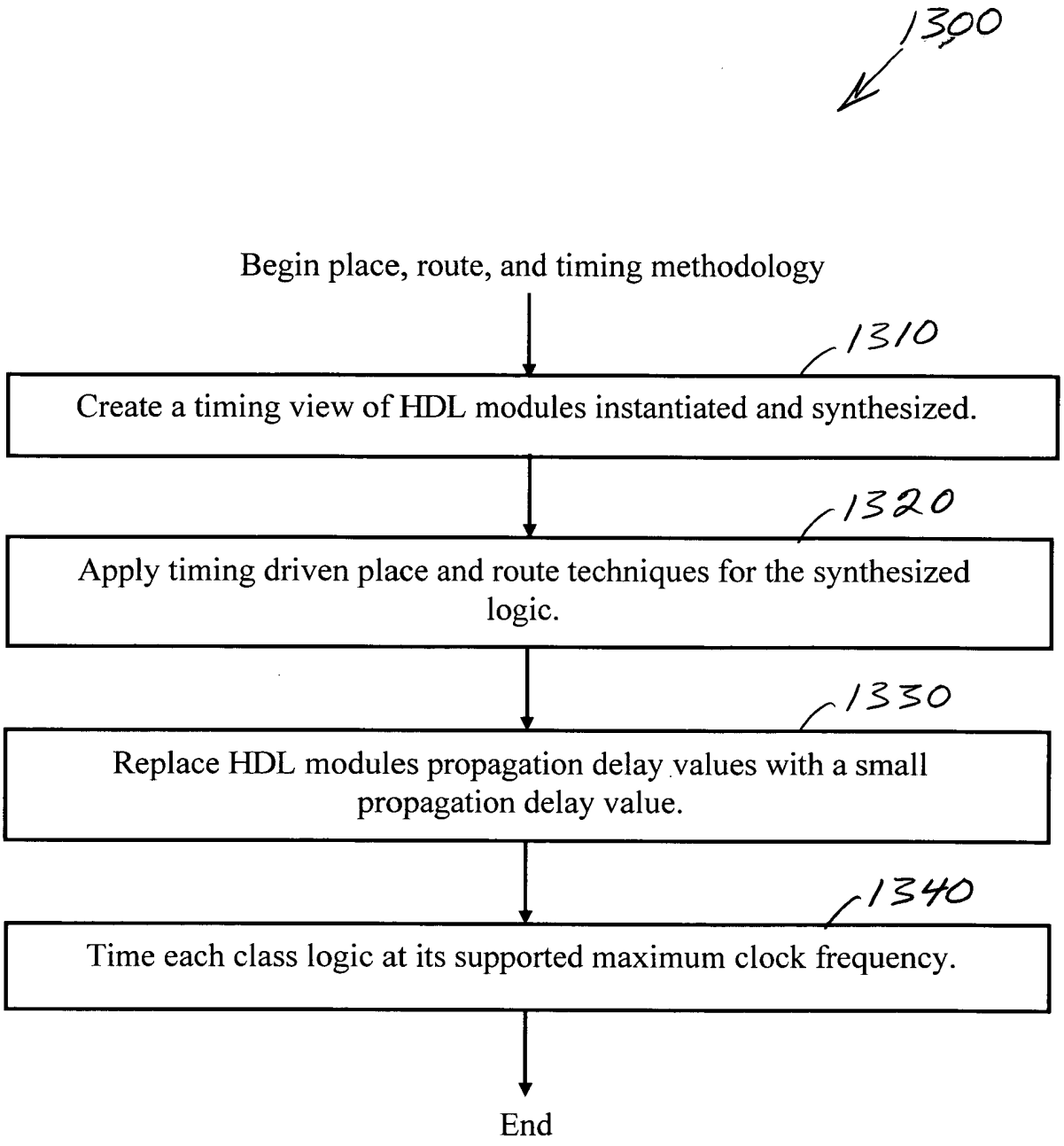

1300
Begin place, route, and timing methodology
1310
Create a timing view of HDL modules instantiated and synthesized.
1320
Apply timing driven place and route techniques for the synthesized logic.
1330
Replace HDL modules propagation delay values with a small propagation delay value.
1340
Time each class logic at its supported maximum clock frequency.
End

Fig. 13

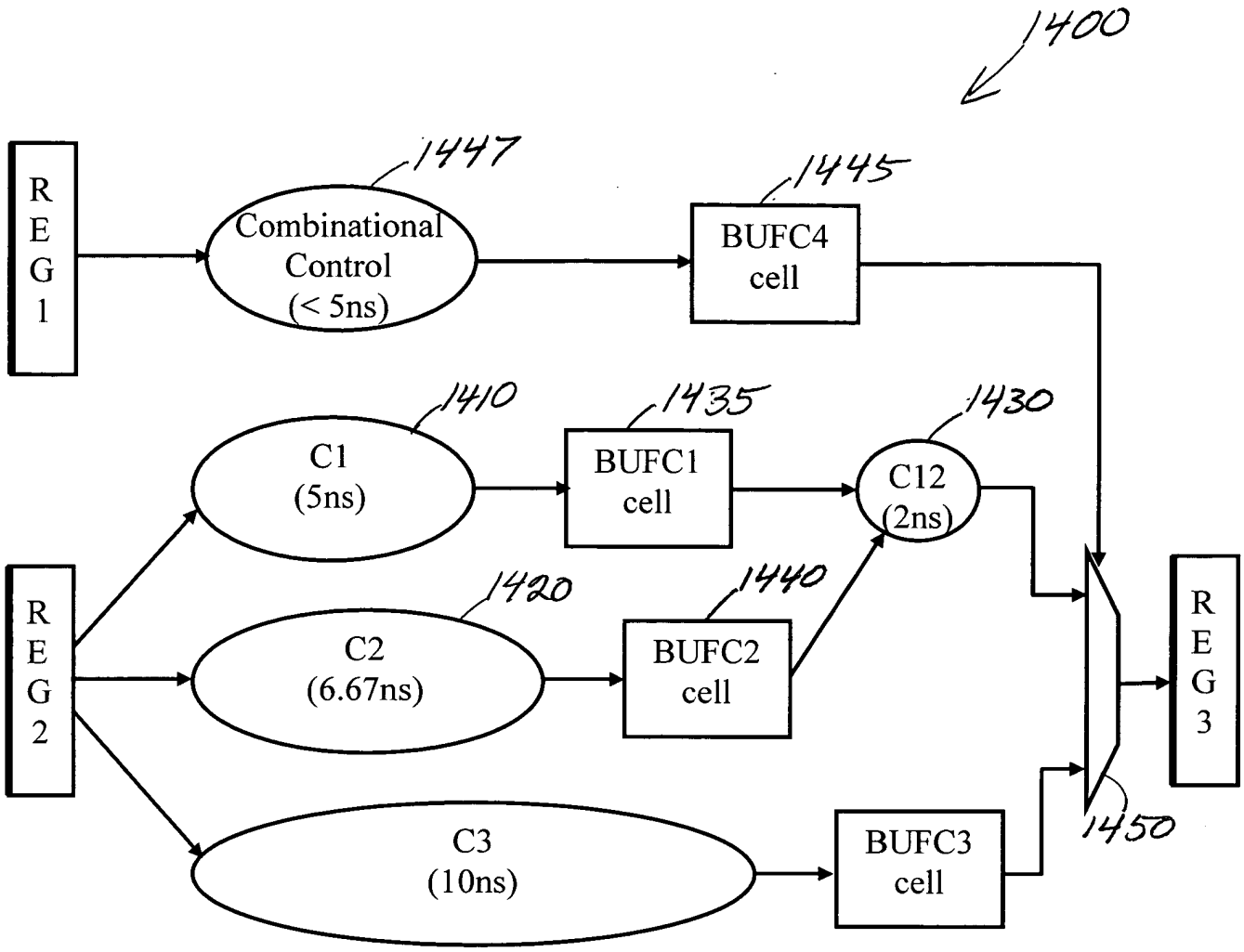

1400
1447
1445
REG 1
Combinational Control (< 5ns)
BUFC4 cell
1410
1435
1430
C1 (5ns)
BUFC1 cell
C12 (2ns)
1420
1440
REG 2
C2 (6.67ns)
BUFC2 cell
REG 3
C3 (10ns)
BUFC3 cell
1450

Fig. 14

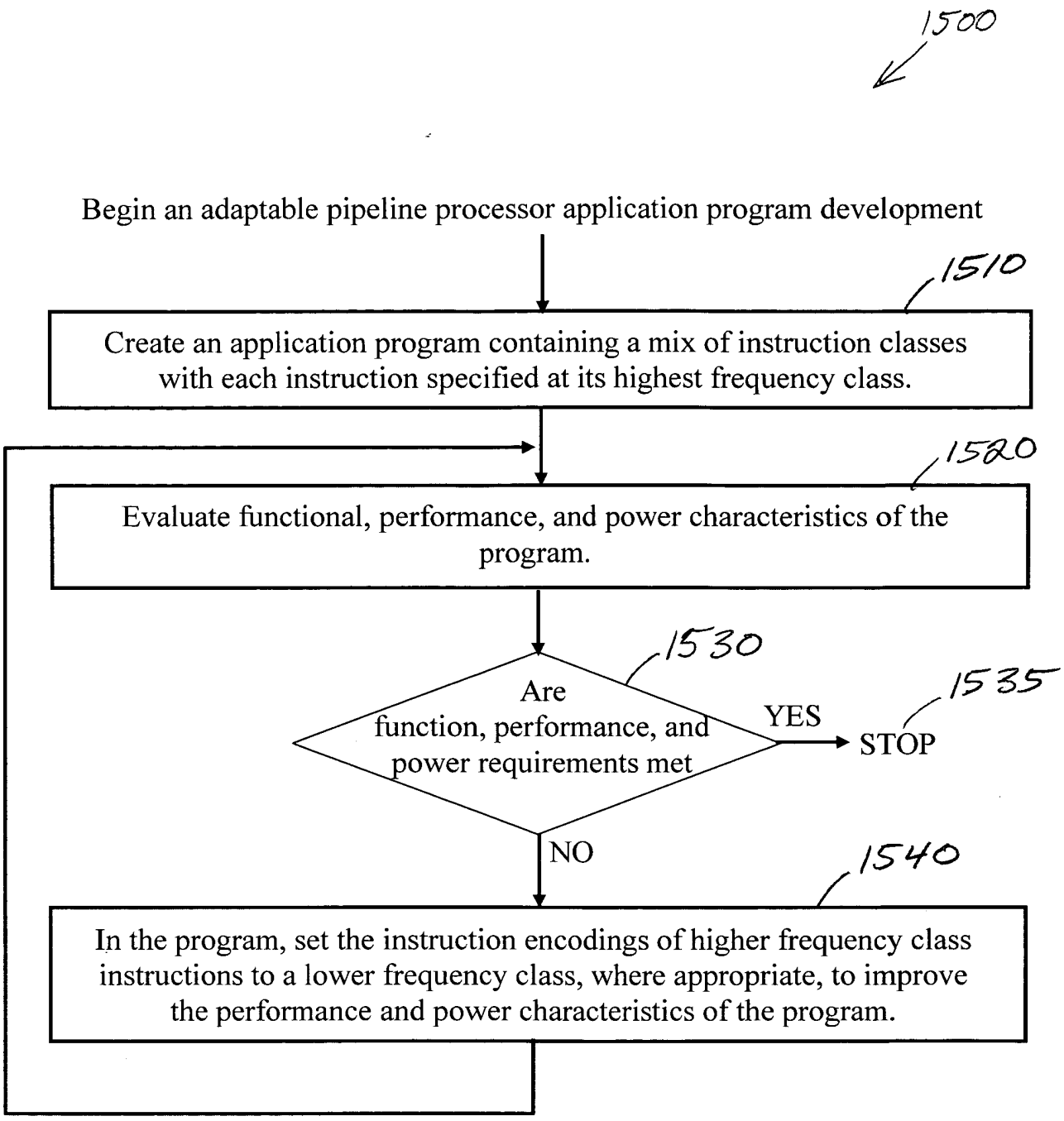

1500
Begin an adaptable pipeline processor application program development
1510
Create an application program containing a mix of instruction classes with each instruction specified at its highest frequency class.
1520
Evaluate functional, performance, and power characteristics of the program.
1530
Are function, performance, and power requirements met
YES
1535
STOP
NO
1540
In the program, set the instruction encodings of higher frequency class instructions to a lower frequency class, where appropriate, to improve the performance and power characteristics of the program.

Fig. 15